SBOLCanvas: A Visual Editor for Genetic Designs

Logan Terry,[†] Jared Earl,[†] Sam Thayer,[†] Samuel Bridge,[†] and Chris J. Myers[*,‡]

[†]*University of Utah, Salt Lake City, 84132, UT, USA*

[‡]*University of Colorado Boulder, Boulder, 80309, CO, USA*

E-mail: chris.myers@colorado.edu

Abstract

SBOLCanvas is a web-based application that can create and edit genetic constructs using the SBOL data and visual standards. SBOLCanvas allows a user to create a genetic design visually and structurally from start to finish. It also allows users to incorporate existing SBOL data from a SynBioHub repository. By the nature of being a web-based application, SBOLCanvas is readily accessible and easy to use. A live version of the latest release can be found at `https://sbolcanvas.org`.

Keywords

Genetic design tool, SBOL, SBOL visual

Introduction

Due to their complicated nature, synthetic biologists often use diagrams to visualize the structure and functionality of genetic designs. The *Synthetic Biology Open Language Visual* (SBOLv)[1,2] is a standard for these diagrams. This standard provides a set of glyphs for

synthetic biology components and how they can interact. These visual designs also have a complementary data standard, the *Synthetic Biology Open Language* (SBOL),[3,4] which represents the structural and functional information for genetic designs.

When a synthetic biology designer is developing a genetic circuit with SBOL and SBOLv, they have three main requirements: 1) an intuitive way to create and edit visual diagrams, 2) an ability to associate these diagrams with genetic part information, and 3) a means to share their designs with others. There are already some tools to aid in this process. VisBOL[5] renders SBOLv diagrams, but requires an SBOL file to generate the diagram from. It does not, however, provide a means to edit these diagrams or customize spacing or colors. DNAplotlib[6] can generate SBOLv diagrams, and it does provide a means to customize color and style. However, DNAplotlib is a command line program, and is not directly tied to the SBOL data model. The tool that comes closest to our goals is SBOLDesigner,[7] a graphical schematic editor for DNA-level design. This tool has many useful features including the ability to construct a DNA sequence from SBOLv glyphs, import DNA part information from the SynBioHub repository,[8] and share resulting designs by uploading them to SynBioHub. However, SBOLDesigner does not support the latest features in SBOLv version 2.[1] SBOLv 2 allows for the inclusion of non-DNA components (RNAs, proteins, small molecules, etc.), as well as a way of representing interactions between them. Furthermore, SBOLDesigner requires local installation to use.

This paper describes SBOLCanvas, an updated web-based genetic design editor that can create visual diagrams using all the features of SBOLv 2. In addition to features supported by SBOLDesigner (i.e., design and visualization of DNA genetic circuits including combinatorial designs), SBOLCanvas also has the ability to:

- Add non-DNA components to these designs,
- Link components via interactions such as genetic production and inhibition,
- Organize components into modules composed of multiple DNA sequences, non-DNA

components, and interactions,

- Store and share designs using the SynBioHub repository,[8]
- Export images of a design,
- Edit metadata and graphical design attributes, and
- General improvements in usability via a web-based interface.

Therefore, SBOLCanvas provides a new way for synthetic biologists to specify and visualize the structure and function of their designs.

Results

This section describes each of these new features of SBOLCanvas in more detail.

Non-DNA components: The SBOL data standard allows for the addition of non-DNA components to genetic circuit design descriptions. These components may be, for example, RNAs or proteins produced by the DNA circuit or that repress or activate production of other proteins. They may also be small molecules that bind to proteins to form complexes that change their ability to behave as transcription factors (see Figure 1).

Interactions: In the SBOL data standard, components can be functionally linked via interactions. These interactions can represent genetic production such as the *mxiE coding sequence* (CDS) in Figure 1 that when transcribed leads to the production of the MxiE protein. Interactions can also represent transcriptional regulation. For example, genetic inhibition is shown by the interaction between the LacI protein and the pTac promoter in Figure 1.

Module organization: SBOLDesigner supports a genetic design composed of just a single DNA sequence component. The SBOL data standard, however, supports modules that group together multiple functional components that act together to perform a desired function like the LacI Producer module in Figure 1. Therefore, SBOLCanvas allows users to

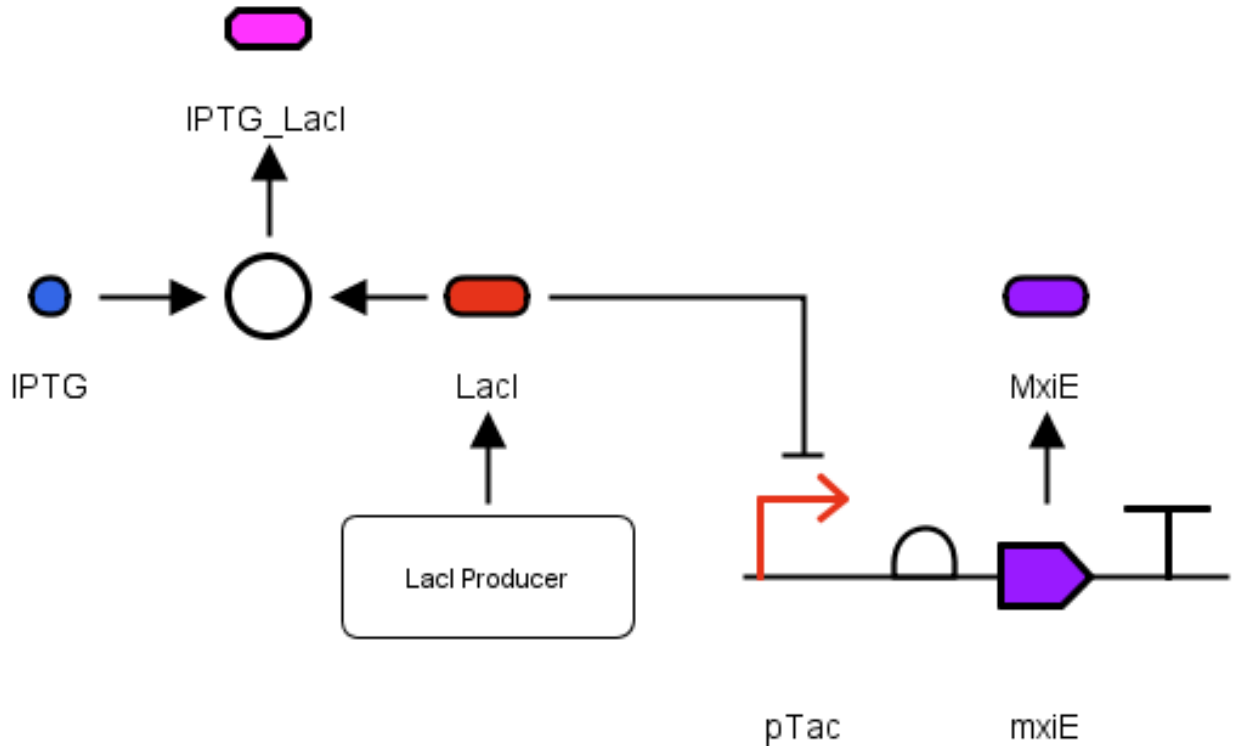

Figure 1: An image exported from SBOLCanvas showing interactions between components and modules in a genetic circuit. In particular, this diagram indicates that the *LacI Producer* module has parts that code for the LacI protein that inhibits the pTac promoter. Another example interaction shown is that the IPTG small molecule binds to the LacI protein to form the IPTG_LacI complex. This interaction sequesters the LacI protein so that it cannot inhibit the pTac promoter.

create such design modules composed of multiple DNA sequence components. Each DNA sequence component may potentially be provided to the cell on the same or separate plasmids. These design modules can also include non-DNA components, as well as interactions between the components.

SynBioHub integration: SynBioHub is a repository for storing genetic design information encoded using the SBOL data standard.[8] SBOLCanvas allows parts to be imported from genetic libraries stored in SynBioHub repositories. Genetic designs created by SBOLCanvas can also be stored in SynBioHub for later access or to share with others.

Image export: If a researcher wants to build and share a diagram of their biological circuit, they often have to use some secondary tool or graphical editor. This is inconvenient

as the secondary tool may not be intuitive or create the diagram exactly as the researcher wants, and creating a diagram with a graphical tool requires manual placement of many lines, and meticulous adjustments. This takes significantly more time than necessary. Since SBOLCanvas is optimized for drawing genetic diagrams and allows users to export images of their designs in a variety of formats (i.e., PNG, JPEG, SVG, and GIF), it is a powerful genetic circuit diagramming tool.

Editing metadata and graphical design attributes: For each element in the SBOLCanvas schematic, the user has access to an *Info* and *Design* menu (see Figure 2). The Info menu allows users to set various metadata, such as the type of part and its identifier, name, and description. The information for a component can be manually entered or imported from a SynBioHub repository. The Design menu allows the user to change graphical attributes, such as stroke and fill color, as well as font color and size.

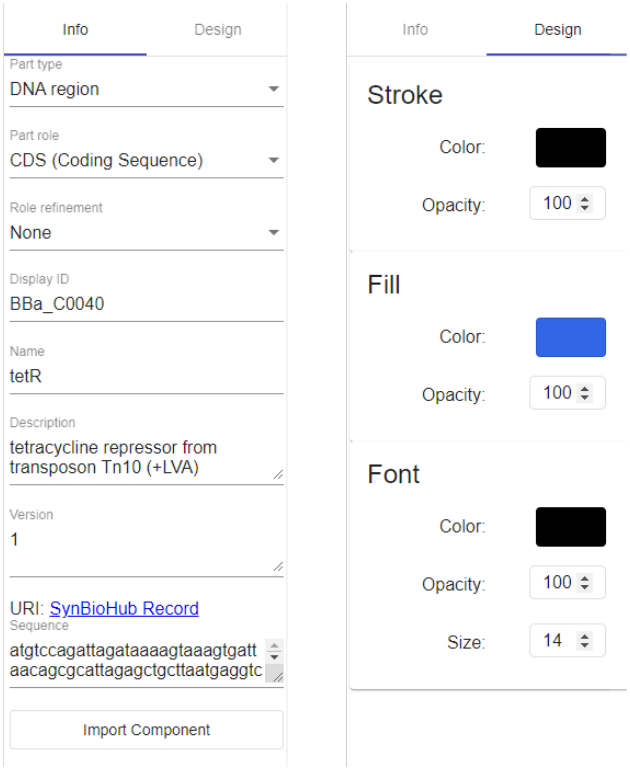

Figure 2: The Info menu on the left provides a means to edit metadata like the part type, and its identifier, name, and description. The Design menu on the right provides a means to change graphical attributes such as the stroke and fill color, as well as font color and size.

Intuitive web-based interface: SBOLCanvas lowers the barrier to entry by enabling anyone with access to a web browser to visually design synthetic genetic circuits. SBOLCanvas' web-based *graphical user interface* (GUI) is shown in Figure 3. The primary way to interact with SBOLCanvas is to drag and drop items from the glyph menu on the left. If a DNA sequence backbone is selected, the user can also click on a glyph to have it added to the end of that DNA sequence backbone. If a user is having trouble scanning for a component they need, they can use the search bar at the top of the glyph menu to filter components. If a user needs to reorder glyphs, they can do so by dragging and dropping the glyph into position. SBOL enables reuse of parts within a design. SBOLCanvas exposes this feature by coupling glyphs. This is done by duplicating their displayIds (an identifier for a glyph) in the Info menu. A prompt helps to determine if the currently selected part, or the conflicting part information should be kept. Any further edits to coupled glyphs will be reflected in their counterparts. Interactions can be added by selecting two items to link, and then clicking on an interaction arrow from the glyph menu. Alternatively, the ends of existing interactions can be moved to link parts. If either end is a Module, a popup appears, prompting selection of a functional component for that end of the interaction. Designs can be stored and shared by uploading them to SynBioHub collections. Designs can be edited or instantiated in other designs by downloading them from SynBioHub. SBOLCanvas also supports exporting the designs to local files in a variety of formats (e.g. SBOL1,[4] SBOL2,[3] GenBank, GFF3, and FASTA).

Methods

SBOLCanvas is architecturally split into two main pieces: the front-end and the back-end. The front-end of SBOLCanvas is built with Angular[9] and MxGraph.[10] The front-end handles all of the user input and manipulates the MxGraph data model while the user builds a construct graphically. The structure of the MxGraph model has been designed in such a

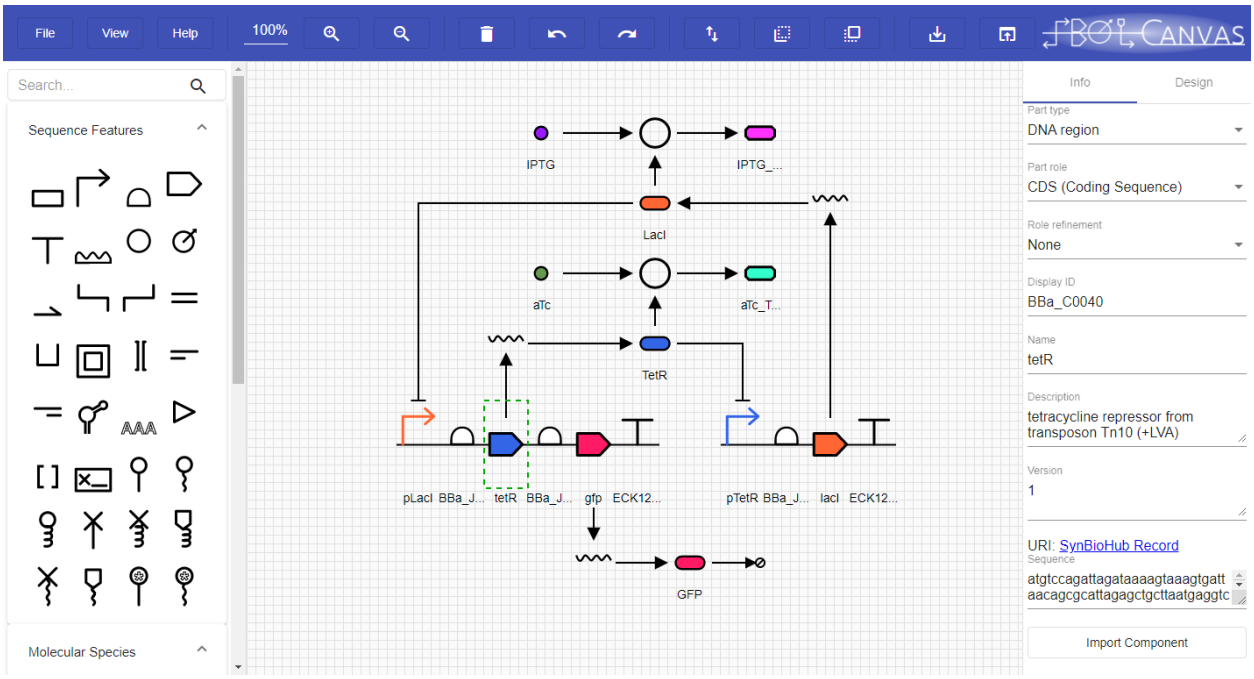

Figure 3: SBOLCanvas' graphical user interface. Main editing canvas in the center. Glyphs can be dragged in from the glyph menu on the left. Metadata and graphical attributes can be changed in the menus on the right. A live version can be found at https://sbolcanvas.org.

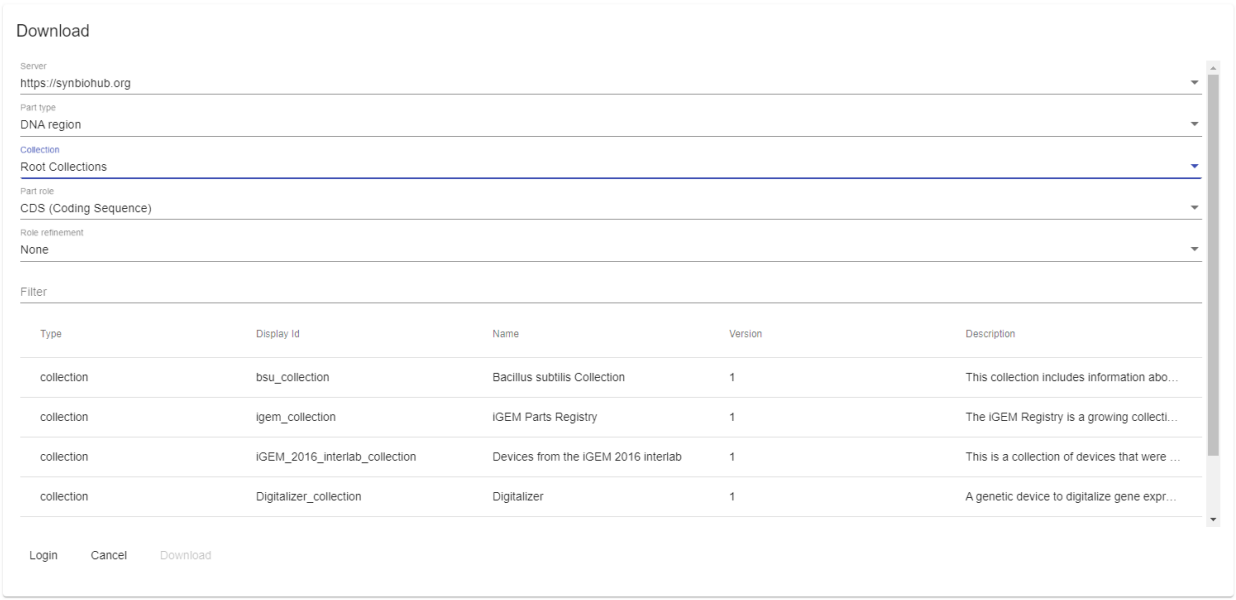

Figure 4: Import from SynBioHub popup with options to refine a search.

way that it closely resembles the SBOL data model. More detail on this can be seen on our GitHub wiki. This makes translation to SBOL easier, as well as enabling the addition of new

features straightforward. The back-end of SBOLCanvas is built with Java, and primarily acts as a converter between SBOL and MxGraph data structures. The SBOL and SynBioHub support is provided by libSBOLj.[11]

SBOLCanvas leverages a strong visual diagramming library, MxGraph, that supports the needed features. As is the case with most libraries, there tends to be project integration issues. The first obstacle was that MxGraph requires certain initialization, which is not well documented, before it could encode and decode the graph model. The encoding and decoding of the MxGraph model is necessary to pass the model to and from the back-end for conversion to SBOL. Another obstacle faced was the architecting of the MxGraph structure. MxGraph organizes its diagrams as a tree of graphical cells, for instance a sequence feature glyph. Originally, our plan was to assume that SBOL followed a tree structure. This made the MxGraph structure quite easy to use as cells already had a parent child relationship. There is also a convenient ability to expand and collapse cells to view their sub-part information. An example of this is using one terminator with sub-part information to represent a double terminator. This structure though had to change to represent the directed acyclic graph structure that SBOL uses. It was no longer possible to use the parent/child relationship to map directly to SBOL. This meant that our tool could not use the expand collapse function in its current form. In order to enable multiple glyphs referencing the same sub-part information, a "view cell" idea had to be adopted. In this model, there is one root cell in the graph, with all the views as it's children. Each glyph then contains a reference to it's corresponding view. When entering a glyph to see its sub-part information, the reference is used to find the corresponding view, so it can be displayed. More info on the MxGraph structure can be found on our GitHub wiki. More info on the SBOL structure can be found at `https://sbolstandard.org`.

Discussion

Since SBOLCanvas was built with a new code base, we endeavored to make it easy to add new features and adapt to changes in currently used libraries. One of the benefits of using the MxGraph structure, as opposed to an SBOL structure, in the front-end is that when the SBOL Version 3[12] Java library is developed, very little of the code base needs to change for SBOLCanvas to support it. As such, SBOLCanvas may be among the first tools to support SBOL 3.

SBOLCanvas is under active development, and there are some enhancements that are planned in the near term. We would like to have a tighter integration with SynBioHub, such as being able to open designs directly from SynBioHub without having to use the download dialog of SBOLCanvas. We also plan to allow users to import collections into the part menu, enabling the ability to drag and drop imported parts into a design. Finally, we would like to support sequence-based search within SBOLCanvas, as well as other general useful search features.

In the future, we plan to have support for automated addition of sequence tags and scars for a variety of assembly methods. When physically constructing designs, certain assembly methods require tags or introduce scars between connected parts. The assembly method used determines the sequence of these tags and scars. We intend to let users choose an assembly method for a design, and add tags or scars to their sequences automatically.

We also plan to give users more of a persistent store of preferences and information. Currently, whenever the page is refreshed, any login and design information is lost. Also, whenever a user goes to download or import a part, the previous collection they were in is forgotten. Another benefit of this persistent state would be a preferred default color or style pallet for glyphs.

In the future, we will be making use of parametric *scalable vector graphics* (SVG) to allow more glyph customization. Each parametric glyph provides certain parameters. Some are more generic, such as width and height, but others can be more specific, like the length

of the promoter arrow. Along with this addition, we hope to make it possible for a user to upload an SVG as their preferred style of a glyph.

Finally, we are currently working on an SBOL layout standard. The intention of this layout standard is to facilitate more uniform rendering across different applications. The initial use of this layout is for exchange with VisBOL,[5] with hopefully other tools soon to follow.

To request a feature or report an issue, please visit:
https://github.com/SynBioDex/SBOLCanvas/issues.

Acknowledgement

SBOLCanvas is supported in part by the SBOL Industrial Consortium. This work is also supported by the National Science Foundation under grant 1939892. An instance of SBOLCanvas (`https://sbolcanvas.org`) is hosted on an Azure server provided by Microsoft Research. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the funding agencies.

Author Contributions

L.T., J.E., and S.T. worked on SBOLCanvas. S.B. created the docker workflow. C.J.M. manages the SBOLCanvas project. L.T., J.E., S.T., and C.J.M. contributed to the writing of this manuscript.

Conflicts of Interest

The authors declare no conflicts of interest.

References

(1) Beal, J.; Nguyen, T.; Gorochowski, T. E.; Goñi-Moreno, A.; Scott-Brown, J.; McLaughlin, J. A.; Madsen, C.; Aleritsch, B.; Bartley, B.; Bhakta, S. et al. Communicating Structure and Function in Synthetic Biology Diagrams. *ACS Synthetic Biology* **2019**, *8*, 1818–1825, PMID: 31348656.

(2) Quinn, J. Y.; Cox, R. S., III; Adler, A.; Beal, J.; Bhatia, S.; Cai, Y.; Chen, J.; Clancy, K.; Galdzicki, M.; Hillson, N. J. et al. SBOL Visual: A Graphical Language for Genetic Designs. *PLOS Biology* **2015**, *13*, 1–9.

(3) Roehner, N.; Beal, J.; Clancy, K.; Bartley, B.; Misirli, G.; Grünberg, R.; Oberortner, E.; Pocock, M.; Bissell, M.; Madsen, C. et al. Sharing Structure and Function in Biological Design with SBOL 2.0. *ACS Synthetic Biology* **2016**, *5*, 498–506, PMID: 27111421.

(4) Galdzicki, M.; Clancy, K. P.; Oberortner, E.; Pocock, M.; Quinn, J. Y.; Rodriguez, C. A.; Roehner, N.; Wilson, M. L.; Adam, L.; Anderson, J. C. et al. The Synthetic Biology Open Language (SBOL) provides a community standard for communicating designs in synthetic biology. *Nature Biotechnology* **2014**, *32*, 545–550.

(5) McLaughlin, J. A.; Pocock, M.; Mısırlı, G.; Madsen, C.; Wipat, A. VisBOL: Web-Based Tools for Synthetic Biology Design Visualization. *ACS Synthetic Biology* **2016**, *5*, 874–876, PMID: 26808703.

(6) Der, B. S.; Glassey, E.; Bartley, B. A.; Enghuus, C.; Goodman, D. B.; Gordon, D. B.; Voigt, C. A.; Gorochowski, T. E. DNAplotlib: Programmable Visualization of Genetic Designs and Associated Data. *ACS Synthetic Biology* **2017**, *6*, 1115–1119, PMID: 27744689.

(7) Zhang, M.; McLaughlin, J. A.; Wipat, A.; Myers, C. J. SBOLDesigner 2: An Intuitive Tool for Structural Genetic Design. *ACS Synthetic Biology* **2017**, *6*, 1150–1160, PMID: 28441476.

(8) McLaughlin, J. A.; Myers, C. J.; Zundel, Z.; Mısırlı, G.; Zhang, M.; Ofiteru, I. D.; Goñi-Moreno, A.; Wipat, A. SynBioHub: A Standards-Enabled Design Repository for Synthetic Biology. *ACS Synthetic Biology* **2018**, *7*, 682–688, PMID: 29316788.

(9) Agius, A.; Rickabaugh, A.; Kushnir, A.; Scott, A.; Seguin, A.; Wang, A.; Lyding, C.; Greene-Kaplan, C.; Shevitz, D.; Parker, D. et al. Angular. 2016; `https://angular.io/`.

(10) Alder, G.; Benson, D. MxGraph. 2005; `https://github.com/jgraph/mxgraph`.

(11) Zhang, Z.; Nguyen, T.; Roehner, N.; Mısırlı, G.; Pocock, M. R.; Oberortner, E.; Samineni, M.; Zundel, Z.; Beal, J.; Clancy, K. et al. libSBOLj 2.0: A Java Library to Support SBOL 2.0. *IEEE Life Sciences Letters* **2015**, *1*, 34–37.

(12) McLaughlin, J. A.; Beal, J.; Mısırlı, G.; Grünberg, R.; Bartley, B. A.; Scott-Brown, J.; Vaidyanathan, P.; Fontanarrosa, P.; Oberortner, E.; Wipat, A. et al. The Synthetic Biology Open Language (SBOL) Version 3: Simplified Data Exchange for Bioengineering. *Frontiers in Bioengineering and Biotechnology* **2020**, *8*.

Graphical TOC Entry